

THIS FILE COPY

WHOI-89-13

②

AD-A210 278

# Woods Hole Oceanographic Institution



## Satellite Data Processing System (SDPS) Users Manual V1.0

by

Michael Caruso  
Chris Dunn

May 1989

### Technical Report

Funding was provided by the Office of Naval Research under contract Number N00014-86-K-0751; and by the National Aeronautic and Space Administration under contract Number 957652.

Approved for public release; distribution unlimited.

DTIC  
ELECTE  
JUL 18 1989  
S E D

89 7 17 128

WHOI-89-13

**Satellite Data Processing System (SDPS)  
Users Manual V1.0**

by

Michael Caruso  
Chris Dunn

Woods Hole Oceanographic Institution  
Woods Hole, Massachusetts 02543

May 1989

**Technical Report**

Funding was provided by the Office of Naval Research  
under contract Number N00014-86-K-0751; and by the National Aeronautic  
and Space Administration under contract No. 957652.

Reproduction in whole or in part is permitted for any purpose of the  
United States Government. This report should be cited as:  
Woods Hole Oceanog. Inst. Tech. Rept., WHOI-89-13.

Approved for publication; distribution unlimited.

**Approved for Distribution:**

A handwritten signature in dark ink, reading "Robert C. Beardsley", is written over a horizontal line.

**Robert C. Beardsley, Chairman**  
Department of Physical Oceanography

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>System Requirements</b>	<b>1</b>
<b>3</b>	<b>Conventions</b>	<b>1</b>
<b>4</b>	<b>Window Organization</b>	<b>2</b>
4.1	Menu Window	2
4.2	Display Window	4
4.2.1	Video Lookup Tables (VLTs)	4
4.3	Cursor Window	5
<b>5</b>	<b>Filetypes</b>	<b>5</b>
5.1	ASCII Files	6
5.2	Binary Files	6
5.3	Sun Rasterfiles	6
5.4	SDPS Format Files	6
<b>6</b>	<b>Basic Image Display</b>	<b>7</b>
6.1	Window and Memory Numbering	10
6.2	Using the Cursor Window	12
<b>7</b>	<b>Basic Line Display</b>	<b>14</b>
7.1	Window Scaling	14
7.2	Converting data to SDPS format	16
7.3	Lines	17
7.3.1	Additional line types	18
7.4	Vectors	18
<b>8</b>	<b>Basic Image Enhancement</b>	<b>21</b>
8.1	VLT Functions	21
8.1.1	Colorbar	23
8.1.2	Cursor Controlled Linear Mapping (CCLM)	25
8.2	Image Functions	25
8.2.1	Histogram	25
8.2.2	Zoom	25

8.2.3	Math . . . . .	28
8.3	Text . . . . .	30
<b>9</b>	<b>Basic Output</b>	<b>32</b>
9.1	SDPS Format . . . . .	32
9.2	PostScript . . . . .	33
9.3	UNIX . . . . .	33
<b>10</b>	<b>Advanced Functions</b>	<b>33</b>
10.1	VLT Modification . . . . .	34
10.2	Movie Loop . . . . .	34
10.3	Plotting Scalar Values Along a Track . . . . .	36
10.4	Line Drawing . . . . .	38
10.5	Bitmap overlays . . . . .	38
10.6	Digitize Points . . . . .	39
10.7	Dithered Images . . . . .	39
10.8	Defaults . . . . .	39
<b>11</b>	<b>Menu Listings and Descriptions</b>	<b>39</b>
11.1	Window Functions . . . . .	39
11.2	VLT Manipulation . . . . .	41
11.3	Raster Image Display . . . . .	43
11.4	Track Data Plotting . . . . .	44
11.5	File Reformatting . . . . .	47
11.6	Data Processing Functions . . . . .	47
11.7	Dither And PostScript Functions . . . . .	48
11.8	Image Functions . . . . .	48
<b>12</b>	<b>External Utilities</b>	<b>49</b>
12.1	sdps_append . . . . .	49
12.2	sdps_convert . . . . .	50
12.3	sdps_cut . . . . .	51
12.4	sdps_flip . . . . .	51
12.5	sdps_ftb . . . . .	52
12.6	sdps_header . . . . .	53
12.7	sdps_headedit . . . . .	53
12.8	sdps_logscale . . . . .	53

12.9	sdps_mask . . . . .	53
12.10	sdps_math . . . . .	54
12.11	sdps_pad . . . . .	54
12.12	sdps_rotate . . . . .	55
12.13	sdps_to_sun . . . . .	55
<b>A</b>	<b>Example ASCII Line data file</b>	<b>57</b>
<b>B</b>	<b>Example ASCII Vector data file</b>	<b>57</b>

## List of Figures

1	SDPS with one open display window. . . . .	3
2	Example of an image printed on a PostScript laser printer. Image courtesy of NORDA. . . . .	11
3	Example of plotting a continuous line. Grid lines were added using option 42 and notation was added using option 46 . . . . .	19
4	Example of a vector plot. Box was added using option 42 and notation was added using option 46. . . . .	22
5	Example of a scalar plot. Coastline was added as a discontinuous line using option 43. . . . .	37

## 1 Introduction

SDPS is a menu driven interactive program designed to facilitate the display and output of image and line-based data sets common to telemetry, modeling and remote sensing. This program can be used to display up to four separate raster images and overlay line-based data such as coastlines, ship tracks and velocity vectors. The program uses multiple windows to communicate information with the user. At any given time, the program may have up to four image display windows as well as auxiliary windows containing information about each image displayed. SDPS is not a commercial program. It does not contain complete type checking or error diagnostics which may allow the program to crash. Known anomalies will be mentioned in the appropriate section as notes or cautions.

## 2 System Requirements

SDPS was designed to be used on Sun\* Microsystems Workstations running SunView1 (Sun Visual/Integrated Environment for Workstations). It was primarily designed to be used on workstations equipped with color monitors, but most of the line-based functions and several of the raster-based functions can be used with monochrome monitors. The program currently runs on Sun3 series workstations running SunOS 4.0 and should port easily to Sun4 and Sun386 series workstations with SunView1. Users should also be familiar with UNIX<sup>†</sup>, Sun workstations and the SunView window system.

## 3 Conventions

All window information is given in pixels except where the program requests degrees. Pixels are assumed to be integers while degrees are decimal degrees. File names are assumed to be complete with extensions. In the text the following conventions are used: *.bin* for unformatted binary; *.sdps* for SDPS type byte; *.sdpsf* for SDPS type float; and *.asc* for ASCII files<sup>‡</sup>.

---

\*Sun, Sun3, Sun4, Sun386 and SunView1 are trademarks of Sun Microsystems

†UNIX is a trademark of AT&T Bell Laboratories

‡See section 5.4 for more information on file types

## 4 Window Organization

SDPS is a windows-based program consisting of three basic windows (figure 1). The menu window used for input and output, the display window for lines and images and the cursor window for cursor information.

### 4.1 Menu Window

When SDPS is initiated by typing `sdps`<sup>2</sup>, a menu window is opened showing the following main menu:

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 10 Window functions                   :
: 20 VLT Manipulation                   :
: 30 Raster/vector image display        :
: 40 Track data plotting                 :
: 50 File formatting                    :
: 60 Data processing functions          :
: 70 Dithering functions                 :
: 80 Image functions                    :
: 99 Exit program                       :
:      Type <ESC> to abort function.    :
:-----:
```

This window is used for text input and output by the user. This menu shows the list of available submenus. These additional menus are accessed by entering the appropriate number and pressing return. These commands will be described in subsequent sections. **Note:** if you type something incorrectly in this window, you can abort that function and return to the submenu by hitting the *escape key* <ESC>. Also, typing a *carriage return* <CR> in any submenu will return you to the main menu. You can get to any function by simply typing the number of the function. You do not have to return to the main menu and enter each submenu. This is done in the basic examples to familiarize the reader with the program organization.

---

<sup>2</sup>See your system manager if you have problems starting the program with this command



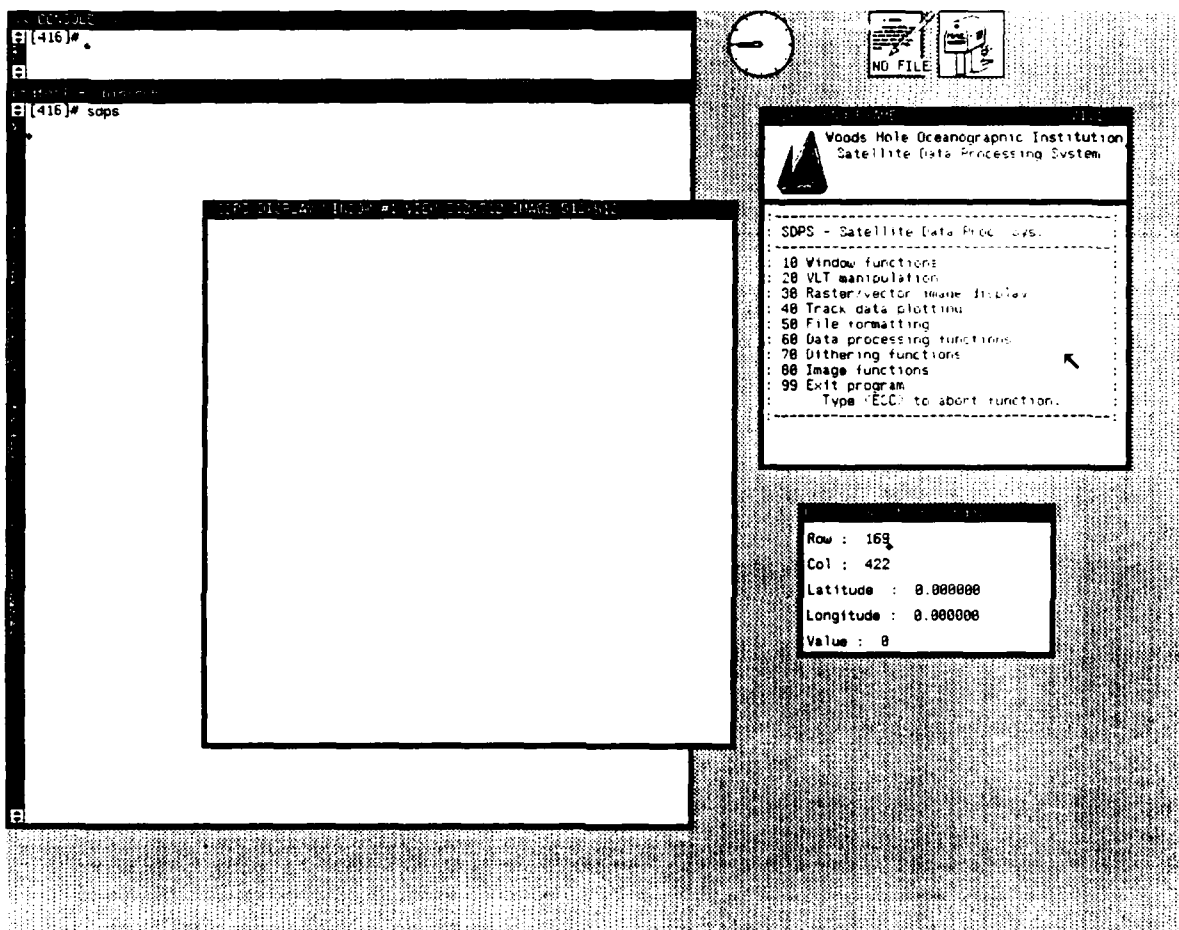


Figure 1: SDPS with one open display window.

## 4.2 Display Window

This window is used for graphic input and output. All display windows start with the same configuration. Although there are many different terms for the coordinates of an image, SDPS uses *rows* and *columns*. *Rows* increase from top to bottom and *columns* increase from left to right. The origin of an image is located in the upper left hand corner at row 0 and column 0. The display window may be opened to any size desired by the user<sup>3</sup>. If a window is larger than the screen size, the user may specify one value for the view window and a separate, larger value for the image size (window and image sizes are given in pixels). Therefore, if you wanted to display a 1536 by 1536 image on an 1152 by 900 Sun monitor, you could specify a view window of 700 by 700 and an image size of 1536 by 1536. You would get a display window that was 700 by 700 pixels in size with scroll bars<sup>4</sup> on it. A maximum of four open display windows is allowed. If the user tries to open more windows, an error message is printed and the program returns to the main menu.

### 4.2.1 Video Lookup Tables (VLTs)

SDPS was designed to work primarily with 8-bit color workstations. This will only allow the program to display 256 individual colors at one time. However, there are more than 16 million possible colors to choose from to display. Therefore, a VLT is used to map up to 256 of the available 16 million colors. Each open display window may have its own VLT which becomes active when the cursor is placed in the window. Because all 256 colors are active in the display window, the foreground and background colors used in the text windows are temporarily switched out while the display colors are switched in. This may be disconcerting since the only window visible is the display window. Information on how to change this effect will be given later. When using the program with a color or greyscale monitor, each display window is opened and given a linear greyscale VLT. The colors in this map run from pure black to pure white.

---

<sup>3</sup>Practical limits vary from 2048 by 2048 to 4096 by 4096 depending on system configuration

<sup>4</sup>See Sunview Beginners Guide

### 4.3 Cursor Window

This window is also opened when the program starts and should not be removed. The window may be moved or closed if the user desires.

```
Row   : 0
Col   : 0
Latitude : 0.000000
Longitude : 0.000000
Value  : 0
```

This window is used to display continuous information to the user. When the cursor enters a display window, it changes to a cross hair and the cursor window gives information specific to that window. The row and column numbers are always updated as is the value number (the original pixel value) under the cursor. The latitude and longitude only change after window specific navigation information has been entered. More detailed information is given in section 6.2.

## 5 Filetypes

Data files to be displayed may be of two distinct file types: image (or raster-based) and vector (or line-based). An image file is a two dimensional function  $\mathcal{F}(x,y)$ . When displaying an image visually, this function is represented by the light intensity displayed. A digital image is the result of spatially discretizing the image function  $\mathcal{F}$ . SDPS currently only supports digital images. All unevenly or irregularly spaced images must be discretized to an even spatial grid. Each dimension may have a different spacing, but it must be constant for that direction.

The line data type contains a series of one or more points. Lines may be continuous, discontinuous, discrete points or vectors. Both data types may be composed of either byte values, integer values or floating point values. SDPS was primarily designed to read and write a specific file format (section 5.4). This format can handle both image and line data types and is in binary format to make reading and writing faster. All data must be converted to SDPS format before using SDPS. The only exceptions to this are Sun rasterfiles (section 5.3) and simple binary images (section 5.2). The idea behind this is that it is better to have a standard format for reading and manipulating images than it is to continually modify code to accept new formats. The SDPS format is given for users who

would like to write conversion programs (section 5.4). Binary images may be read in to allow the user to preview an image before doing any format conversion. Sun rasterfiles may be read in to maintain system compatibility. **Note:** all images must be converted to byte images before using SDPS to display them. However, many external routines (section 12) support additional image types.

## 5.1 ASCII Files

ASCII files are not directly supported by SDPS due to speed considerations. The program does include one option (option 51, section 11.5) as well as an external routine `sdps_convert` (section 12.2) to convert ASCII files. It is recommended that the latter be used since it is a more robust routine.

## 5.2 Binary Files

The only non-SDPS format binary files supported are byte-encoded digital images. These images may have any size header, but any information contained in them is ignored. Binary files may also be converted to SDPS format files using option 52 (section 11.5).

## 5.3 Sun Rasterfiles

The Sun rasterfile format is the basic format for image files used by many Sun workstation programs. The rasterfile is composed of three parts. The first part is the 32 byte header. This header contains 8 integers that described various attributes of the image such as the number of rows, the number of columns and the number of bits per pixel in the image. The second part contains the colormap values. The size of this section is variable from image to image and is given in the header. The third part is the actual image stored row by row. For more information refer to the rasterfile manual page in the file formats section of the Sun manuals.

## 5.4 SDPS Format Files

Most options to SDPS require that input data files be in SDPS format. This is needed since the program requires information when reading the file and often when manipulating the data in memory. This information includes the size, the

type of data, etc. An SDPS format file is simply a binary file with a 512 byte header. The header format is :

SDPS Header Format			
Variable	Type	Bytes	Description
annot	char	80	80 character array used for identifying the image
type	long int	4	type of data: 1 - Byte (Logical*1) 2 - Short integer (Integer*2) 3 - Long integer (Integer*4) 4 - Float (Real*4) 5 - Complex (Complex)
dim	long int	4	number of dimensions in data file (max 4)
ind[4]	long int	16	number of indices for each dimension
inc[4]	float	16	increment between indices in each dimension
slope	float	4	slope of linear relationship between byte and float data
intrcp	float	4	intercept of linear relationship between byte and float data
unused[384]	char	384	fills 512 byte record for data exchange with FORTRAN programs

The data segment of an SDPS format file may be any of the five types listed above, but the two most common formats are byte and float. The other types are included for future expandability. SDPS can read both byte and float files, however image functions currently only support byte type files and the line functions only support float type files. There are several external programs (section 12) that support both types of files.

## 6 Basic Image Display

To display an image using SDPS, the user must know the format and size of the image to be displayed. The format must be either binary, Sun rasterfile or SDPS as described in section (5). The image size must be known in order to open the correct size window. The image size as well as the header size of binary files must

be obtained by the user. The image size for Sun rasterfiles can be obtained from the UNIX command 'file'. The image size for SDPS format files may be determined using the external routine *sdps\_header* (section 12.6). Assuming the file is a binary format file 1024 rows and 768 columns and has a 512 byte header, the first step is to open a window using option 11. Program output is in **Typewriter**, user input is in **Boldface**, comments are in *Italics*:

```

:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 10 Window functions                   :
: 20 VLT Manipulation                   :
: 30 Raster/vector image display        :
: 40 Track data plotting                 :
: 50 File formatting                    :
: 60 Data processing functions          :
: 70 Dithering functions                :
: 80 Image functions                   :
: 99 Exit program                      :
:      Type <ESC> to abort function.    :
:-----:

```

10

*Select the Window Functions Submenu*

```

:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 11 Open window                       :
: 12 Redisplay image                   :
: 13 Flick between images               :
: 14 Output window to memory            :
: 15 Output window to file              :
: 16 Output B+W remapped window to file :
: 17 Clear window                      :
: 18 Display colorbar                   :
: 19 Dump window to unix plot file      :
:      Type <ESC> to abort function.    :
:-----:

```

11 *Select the Open Window Function*

Enter size of view window #1 (row, col)

Default: [512 512]

768 768 *Select a window size in pixels that will fit on the screen*

Enter full image size (row, col): [512 512]

1024 768 *Enter the actual size of the image in pixels*

*A display window will be opened up on the screen 768 pixels high by 768 pixels wide. Also the cursor window will be opened up when you open the first display window. Next load the image into the display window:*

30 *Select the Raster/vector image display submenu*

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 31 Display Sun rasterfile              :
: 32 Display binary image file           :
: 33 Display SDPS image file             :
: 34 Display multiple SDPS image files   :
: 35 Display SDPS mask file              :
: 36 Overlay SDPS bit map                :
:                                         :
:                                         :
:                                         :
:      Type <ESC> to abort function.      :
:-----:
```

32 *Select the binary image display function*

Enter filename of input image ?

file1.bin *Enter the filename of the image*

Enter image memory # and window # [1 1]

1 1                    *Select the first memory location and the window just opened*

Enter display position (row, col) [0 0]

0 0                    *Position it in the upper left hand corner*

Enter image size (row,col) [1024 768]

1024 768              *Enter the image size*

Enter # of bytes of header to skip [0]

512                    *Enter the size of the header to skip*

The image should be loaded into the window with a black and white VLT. The menu will return to the *Raster/image display submenu*. When the cursor enters the display window, the black and white VLT will become active and the image will become visible. The scroll bars can now be used to view the hidden sections of the image. **Note:** most of the scroll bar functions have not been implemented. To move to a different part of the image, the user must use the middle mouse button to move to the section of the image relative to the position of the cursor in the scroll bar. An example of this type of image is given in figure 2. This image was created by NORDA and was generated using option 74. For more information on creating hardcopy of images see section 9. **Note:** User entries are separated by spaces not commas.

## 6.1 Window and Memory Numbering

SDPS supports up to 4 concurrent display windows and up to 50 virtual windows<sup>5</sup>. A virtual window is a copy of a displayable image or window stored in virtual memory and not as disk file. Auxiliary information such as colormaps are not stored along with an image in memory. Therefore memory locations should be used when this information is not needed. For example, the user may want to recall an image of a coastline and not be concerned with knowing the pixel values. The coastline can be quickly re-displayed from memory rather than being read from disk again.

---

<sup>5</sup>Actual amount depends on image sizes and system configuration.



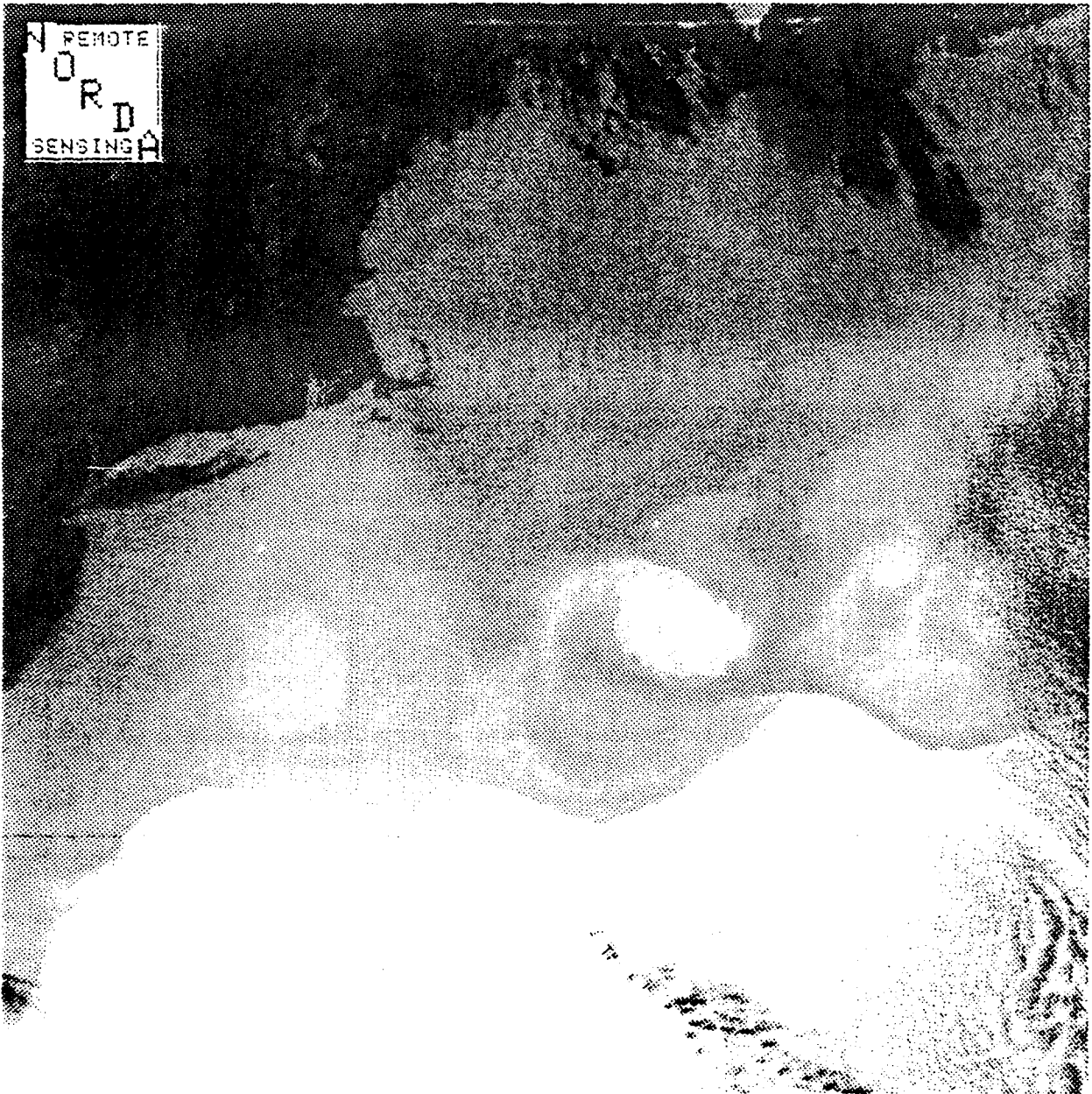


Figure 2: Example of an image printed on a PostScript laser printer. Image courtesy of NORDA.

Each time an image display window is opened, it is numbered from 1 to 4 and its number is shown in the title bar across the top of the window unless the window is too small to display it. If the user opens two windows and then quits window number one, the next window opened will be window number one. A window may be eliminated by selecting the quit option from the window menu.

## 6.2 Using the Cursor Window

The cursor window is opened when a display window is opened. This window gives information about cursor position and pixel values. Since SDPS uses 256 greyscale shades by default (section 4.2.1), the cursor window is not visible when the cursor is in the display window. To make the cursor window visible, the user must truncate the VLT of the display window. This can be done as follows:

```
:-----:
: SDPS - Satelllite Data Proc. Sys.      :
:-----:
: 10 Window functions                    :
: 20 VLT Manipulation                    :
: 30 Raster/vector image display         :
: 40 Track data plotting                  :
: 50 File formatting                     :
: 60 Data processing functions            :
: 70 Dithering functions                  :
: 80 Image functions                     :
: 99 Exit program                        :
:      Type <ESC> to abort function.      :
:-----:
```

20

*Select the VLT Manipulation Submenu*

```

:-----:
: SDPS - Satellite Data Proc. Sys. :
:-----:
: 21 Load BLACK/WHITE VLT :
: 22 Load COLOR VLT :
: 23 Load individual VLT values :
: 24 Read VLT from file :
: 25 Write VLT from file :
: 26 Cursor controlled linear mapping :
: 27 Invert VLT values :
: 28 Set auto-truncation option :
: 29 Set cursor color :
: Type <ESC> to abort function. :
:-----:

```

28 *Select the auto-truncation option*

Auto-truncation flag now set to (0):  
 Auto-truncation will truncate all  
 images to (0-253) range, and  
 maintain VLT values 254=black  
 255=white throughout all functions  
 Enter value(1=ON 0=OFF) ?

1 *Turn on auto-truncation option*  
 21 *Select function to load VLT (21 or 22)*

Enter window # for B+W VLT ?

1 *Select an open window*

Now when the cursor enters the display window, the cursor location will be displayed along with the value of the pixel under the cursor. Activating the latitude/longitude display is demonstrated in section 7.1.

## 7 Basic Line Display

Often it is useful to display various types of lines such as coastlines, wind vectors, ship tracks, satellite ground tracks or buoy information on an image. This section will describe how to accomplish this. Appendix A contains a data set that will be used in the examples given in sections 7.1 and 7.3. Appendix B contains a data set that will be used in the example given in sections 7.4 and 7.2.

### 7.1 Window Scaling

Before any line or vector drawing can be done, the window must be scaled. Since this program was designed for oceanographic data, the windows may be scaled using 3 projections: *Stereographic*, *Mercator* and *Equirectangular*. Although all projections are based on earth coordinates, the equirectangular projection can be given almost any linear coordinates. An example of setting the projection to equirectangular for a flat earth type of projection is shown. The data in appendix A will fit entirely in a latitude/longitude box 25N, 285E to 35N, 295E. First open a window as described in section 6 with a view size of 512 by 512 and an image size of 512 by 512. This function will work with any size window, however the area of interest is square and 512 by 512 is a convenient screen size. **Note:** longitudes should generally be given in east coordinates. West coordinates will work with most functions, but are not fully supported. The user should be cautious when the window spans 0 degrees longitude whether using east or west coordinates.

```

:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 10 Window functions                   :
: 20 VLT Manipulation                   :
: 30 Raster/vector image display        :
: 40 Track data plotting                 :
: 50 File formatting                    :
: 60 Data processing functions          :
: 70 Dithering functions                :
: 80 Image functions                    :
: 99 Exit program                       :
:      Type <ESC> to abort function.    :
:-----:

```

40                                      *Select the track data plotting submenu*

```

:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 41 Set navigation area                 :
: 42 Plot navigation lines               :
: 43 Plot Lat/Lon track                 :
: 44 Plot vectors                       :
: 45 Plot Lat/Lon track with scalar value:
: 46 Text annotation                    :
: 47 Line drawing via cursor control    :
: 48 Digitize points                    :
: 49 Plot contours                      :
:      Type <ESC> to abort function.    :
:-----:

```

41                                      *Select the set navigation area function*

Enter navigation type:

- 1) Stereographic
- 2) Mercator
- 3) Equirectangular

Enter choice?

3                                      *Set the navigation type to equirectangular*

Enter center latitude, longitude ?

30 290                                *Set the latitude and longitude at the center of the window*

Enter rotation angle ?

0                                      *Set the rotation angle to zero*

Enter enter x, y scale factor ?

24 24                                 *Set the column and row scale factors*

Scaling factors are chosen with respect to a 360 degree window. An x scale factor of 1 would give a window scaled +/- 180 degrees from the center longitude. A scale factor of 2 would scale the window +/- 90 degrees etc. The same is true for the latitude or y direction regardless of the window size. If a window is twice as wide as it is high and the user enters a scale factor of 1 1, both the latitude and longitude will be scaled to 360 degrees. Now when the cursor enters the display window, the cursor window will display the location of the cursor in latitude and longitude coordinates as well as row and column values.

## 7.2 Converting data to SDPS format

To make file input faster, SDPS will only read binary files and all non-image files must be in SDPS format. ASCII data files can easily be converted to SDPS format. The ASCII data can be in any format and each point must be separated by a space or a tab. There should not be any extraneous data such as headers or labels and sequential data points should be in columns. For example, the data set in appendix B could be used to plot vectors (option 44, section 11.4) where the first column contains the longitude, the second column the latitude, the third column the u velocity and the fourth column the v velocity. To convert this to SDPS format the user should use the external function *sdps\_convert* (section 12.2):

```
sdps_convert 22 4 < file1.asc > file1.sdpsf
```

The number 22 is the number of rows in the input file and 4 is the number of columns. The user can now display lines and vectors as shown in the following sections.

### 7.3 Lines

This data file can now be plotted in the display window. Assuming that the scaling has been set as in section 7.1 and the user wants to plot a line connecting the positions of each point (e.g. a ship track):

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 41 Set navigation area                 :
: 42 Plot navigation lines               :
: 43 Plot Lat/Lon track                  :
: 44 Plot vectors                       :
: 45 Plot Lat/Lon track with scalar value:
: 46 Text annotation                    :
: 47 Line drawing via cursor control    :
: 48 Digitize points                    :
: 49 Plot contours                       :
:      Type <ESC> to abort function.    :
:-----:
```

43                      *Select the Plot Lat/Lon track function*

Enter window # to use ?

1                      *Set the window number to display data*

Enter SDPS input file name ?

file1.sdpsf            *Enter the filename to read*

Enter indices of lat, lon ?

2 1                    *Column 2 contains the latitude and column 1 contains the longitude*

Enter value to use (0-255) ?

255                    *Enter the VLT value to use in plotting (white)*

Enter type of data set:

- 1) Continuous
- 2) Discontinuous with flag value
- 3) Discon. w/ fixed length data segments
- 4) Individual points

Enter choice ?

1                              *Plot a continuous data set*

A line connecting the data points will now be drawn in the window (figure 3). If the window had an image displayed, the line would have been drawn on top of the image.

#### 7.3.1 Additional line types

SDPS will also read three other types of line data. The most commonly used option is *Discontinuous with flag value*. The data set has a flag such as -9999.0 -9999.0 for the latitude/longitude pair separating continuous line segments. The user can also plot *individual points*. A plus sign is plotted at each coordinate in the data file. Finally, the *Discon. w/ fixed length data segments* option plots multiple line segments each containing the same number of points.

#### 7.4 Vectors

The data file in appendix A can also be used to plot vectors in a display window. Since the data will fit in the same size latitude/longitude box, the scaling should be set the same as in section 7.1. This example will display the data set as vectors with the length proportional to the magnitude (e.g. wind vectors):



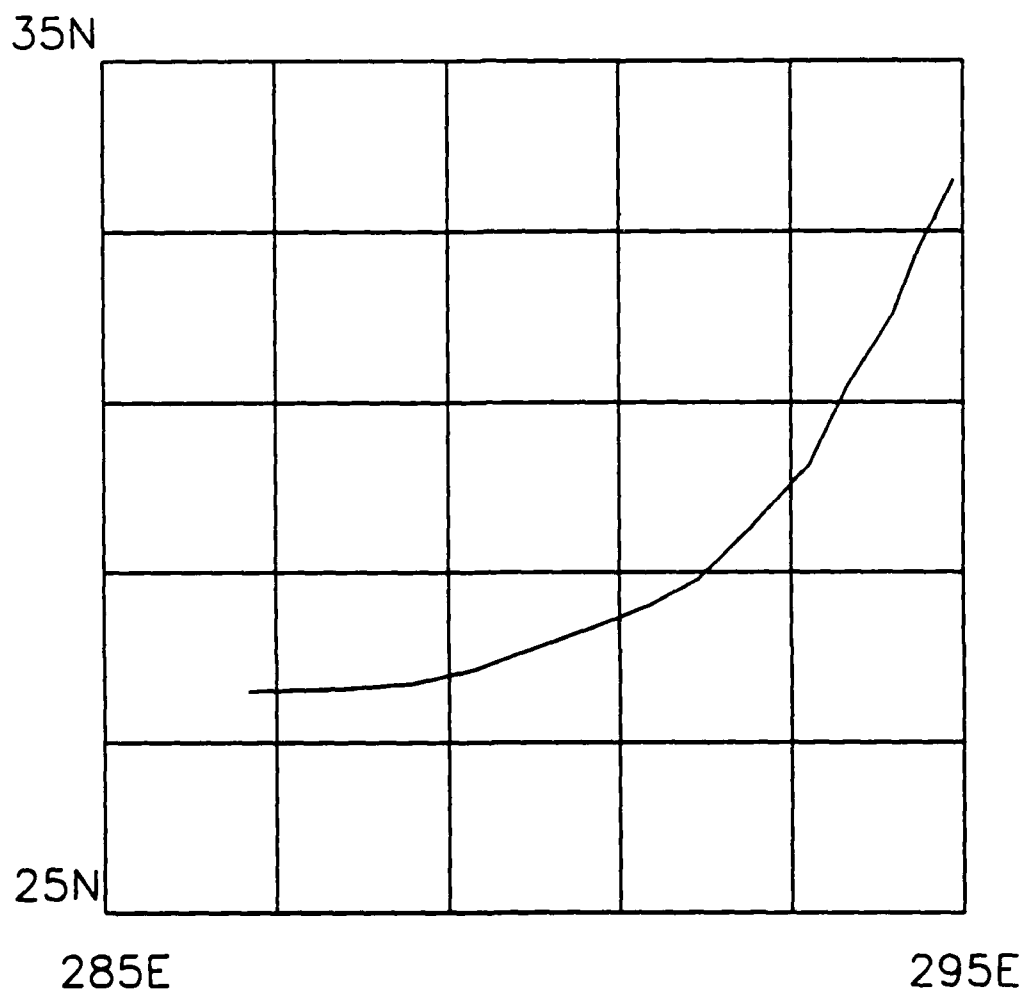


Figure 3: Example of plotting a continuous line. Grid lines were added using option 42 and notation was added using option 46

```

:-----:
: SDPS - Satellite Data Proc. Sys. :
:-----:
: 41 Set navigation area           :
: 42 Plot navigation lines         :
: 43 Plot Lat/Lon track            :
: 44 Plot vectors                  :
: 45 Plot Lat/Lon track with scalar value:
: 46 Text annotation               :
: 47 Line drawing via cursor control :
: 48 Digitize points              :
: 49 Plot contours                 :
:      Type <ESC> to abort function. :
:-----:

```

44                      *Select the plot vectors function*

Enter window # to use ?

1                      *Enter the window number to display vectors*

Enter SDPS input file name ?

file1.sdps              *Set the filename to read*

Enter type of vector data:

1) U,V

2) Magnitude,Direction

Enter choice ?

1                      *Select U,V input data type*

Enter indices of Lat, Lon,U,V ?

2 1 3 4              *Column 2 - lat, column 1 - lon. column 3 - U, column 4 - V*

Enter max expected wind speed ?

7.0                      *Scaling will be done against this value, there is no auto-scale option*

Enter type of plot:

- 1) Vector length varies with magnitude
- 2) Vector value varies with magnitude

Enter choice?

1                      *Select varying vector length option*

Enter length of max  
vector in degrees of longitude ?

2.0                      *A vector with a value of 7.0 will be have a length the same number of pixels as two degrees of longitude*

Enter value to use (0-255) ?

255                      *Enter the VLT value to use in plotting (white)*

Arrows showing the magnitude and direction will now appear in the display window (figure 4). If the display window had an image, the arrows would be drawn on the image. If the type of plot selected is 2, all vectors would be the same length and the color would specify the magnitude.

## 8 Basic Image Enhancement

This section will explain how to modify a display window and to obtain useful information about the data in the window. The user can modify the VLT, view a magnified section of a display, combine information from two windows or add colorbars and text to the display.

### 8.1 VLT Functions

SDPS defaults to a greyscale VLT when the user opens a new window. A user may want to display the current values or to change the VLT to display false colors or to change the mapping of the VLT to the display values.

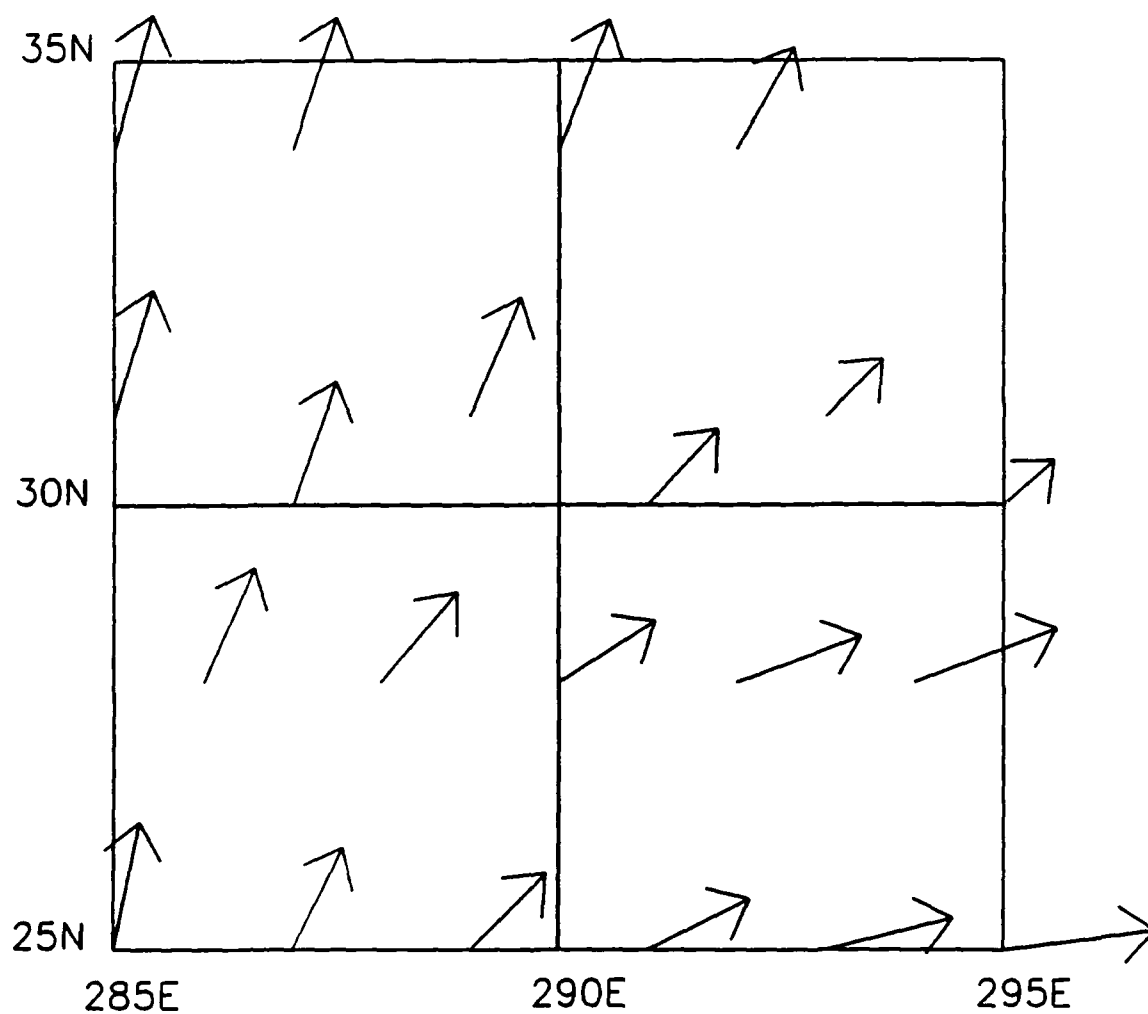


Figure 4: Example of a vector plot. Box was added using option 42 and notation was added using option 46.

### 8.1.1 Colorbar

The current VLT may be displayed in a colorbar in a separate window, or it may be placed in a display window over an image. Assuming the user has opened a window and loaded an image as described above, a colorbar can be overlaid on an image with the following steps:

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 10 Window functions                   :
: 20 VLT Manipulation                   :
: 30 Raster/vector image display        :
: 40 Track data plotting                 :
: 50 File formatting                    :
: 60 Data processing functions          :
: 70 Dithering functions                :
: 80 Image functions                    :
: 99 Exit program                       :
:      Type <ESC> to abort function.    :
:-----:
```

10

*Select VLT Manipulation submenu*

```

:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 11 Open window                        :
: 12 Redisplay image                   :
: 13 Flick between images               :
: 14 Output window to memory            :
: 15 Output window to file              :
: 16 Output B+W remapped window to file :
: 17 Clear window                      :
: 18 Display colorbar                   :
: 19 Dump window to unix plot file      :
:      Type <ESC> to abort function.    :
:-----:

```

18 *Select Display colorbar function*

Enter window # for colorbar ?

1 *Enter the window number*

Place colorbar (0)inside or (1)outside

0 *Specify that the colorbar be placed inside the window*

Enter row, col, height, width:

10 15 20 256 *Enter the starting location, height and width in pixels*

This will put a colorbar in the display window 10 rows from the top and 15 columns from the left. The colorbar will be 20 pixels high and 256 pixels wide. The colorbar can also be put into a separate window by selecting option 18. Enter the window number to read the VLT and enter 1 to specify that the colorbar be placed outside the window. A window in the upper left hand corner of the screen will be created. If the image loaded was an SDPS format file (i.e., if the header contains information on pixel values, section 5.4), the values corresponding to the different colors will be displayed. **Note:** when placing a colorbar inside a window, if the height is greater than the width, a vertical colorbar will be drawn. Otherwise, a horizontal colorbar is drawn. Also colorbars inside windows must be annotated by hand.

### **8.1.2 Cursor Controlled Linear Mapping (CCLM)**

The easiest way to modify the VLT of a window is to use the CCLM option. When this option is selected, the the cursor position controls the VLT map. Moving the cursor vertically changes the contrast; moving it horizontally changes the brightness (black and white) or the color range. To change the VLT, move the cursor and press the left mouse button. The original VLT can be recovered by pressing the cursor one quarter from the top in the middle of the window , or by reloading a VLT (options 21 and 22). The CCLM function is chosen by selecting option 26 and entering the window number to use. Now when the left mouse button is pressed in the window selected, the VLT will be remapped accordingly. The new VLT may be saved using option 25 and restored later using option 24.

## **8.2 Image Functions**

SDPS also provides functions for extracting additional information about images. There are functions to display a histogram of the image, magnify a section of a display window, perform simple math operations on two windows and overlay text.

### **8.2.1 Histogram**

The histogram function reads the pixel values from a specified window and graphically displays the results in a new window. The window shows the number of pixels of each possible value (0-255) on a logarithmic scale. The colors in the histogram correspond to the colors in the display window from which the histogram was computed. A histogram is generated by selecting option 83 and specifying the window number.

### **8.2.2 Zoom**

SDPS also provides a mechanism for magnifying an image (panning is automatically provided with the use of scroll bars). There are two types of magnification available. The user may select a simple pixel replication or a bi-linear interpolation. The replication method is much faster and allows the user to stretch an image by magnifying more in one direction than the other. The interpolation method is slower and smooths edges in the process. Interpolating sharp features such as lines and vectors may produce undesirable effects. Assuming the user has

loaded an image as described above (section 6 ), the following example shows how to magnify a section of the image.

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 10 Window functions                   :
: 20 VLT Manipulation                   :
: 30 Raster/vector image display        :
: 40 Track data plotting                 :
: 50 File formatting                    :
: 60 Data processing functions           :
: 70 Dithering functions                 :
: 80 Image functions                     :
: 99 Exit program                       :
:      Type <ESC> to abort function.     :
:-----:
```

10                                      *Select the Window Functions Submenu*

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 11 Open window                        :
: 12 Redisplay image                    :
: 13 Flick between images                :
: 14 Output window to memory             :
: 15 Output window to file               :
: 16 Output B+W remapped window to file :
: 17 Clear window                       :
: 18 Display colorbar                   :
: 19 Dump window to unix plot file       :
:      Type <ESC> to abort function.     :
:-----:
```

11                                      *Open a new window to hold magnified image*

Enter size of view window #2 (row, col)  
Default: [512 512]



**384 384** *Select a window size for the magnified image*

```
Enter full image size (row, col):[512 512]
```

**384 384**      *Select a view size that will hold the magnified image*

80 *Select the Image functions submenu*[illegible]

**81** *Select the Magnify image function*

Enter window # to read from ?

**1**                      *Select the window to read from*

Enter memory #, window # to write to ?

**2 2**      *Select the memory number and window to put the magnified image*

Enter magnification type?

## 1 - Pixel Replication

## 2 - Bi-linear Interpolation

**1**      *Select simple pixel replication*

Enter magnification factors(row, col) ?

4 4                      *Magnify 4 times in each direction*

Enter window upper left corner (row, col)?

0 0                      *Select the upper left corner*

Enter window lower right corner (row,col)?

95 95                    *Select an area 96 by 96 pixels*

The magnified image will now appear in window number two. The bi-linear interpolation method linearly interpolates between rows and columns to magnify the image. The magnification factor is the same for both directions and must be a power of two.

### 8.2.3 Math

SDPS allows the user to add, subtract, multiply or divide one displayed image by another. These images must be SDPS format images and must be loaded using option 33. The images are first converted back to floating point values using the slope and intercept header values described in section 5.4. Then the selected option is performed and the result transformed back to byte values. The user may choose to keep the original byte mapping or have the program calculate a new slope and intercept. In all operations, the first image is operated on by the second and replaces the second, i.e.,  $image2 = image1 [+ - * /] image2$ . Assuming the user has two windows open and wants to find the difference between the two images, the following steps would accomplish that.



Enter second memory #, window # ?

2 2                      *Select the window to put the result*

Enter function ?

1 - Add

2 - Subtract

3 - Multiply

4 - Divide

2                      *The second image will be subtracted from the first*

Rescale ? 1 - Yes, 0 - No

0                      *Keep the same scale used in the second window*

The function will be performed and the result will be placed in the second window.

### 8.3 Text

Text is also supported in SDPS. All text input and positioning is done by the user. All text is raster-based and will be output the same as other line-based data (see Basic Output section 9). Text style and size may be selected and examples are shown below. All parameters used for text are in pixels so that the window does not have to be scaled (option 41). An example of writing text to a window is shown:

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 10 Window functions                   :
: 20 VLT Manipulation                   :
: 30 Raster/vector image display        :
: 40 Track data plotting                 :
: 50 File formatting                    :
: 60 Data processing functions           :
: 70 Dithering functions                 :
: 80 Image functions                     :
: 99 Exit program                       :
:      Type <ESC> to abort function.    :
:-----:
```

40

*Select the Track data plotting submenu*

```
:-----:
: SDPS - Satellite Data Proc. Sys.      :
:-----:
: 41 Set navigation area                 :
: 42 Plot navigation lines               :
: 43 Plot Lat/Lon track                  :
: 44 Plot vectors                       :
: 45 Plot Lat/Lon track with scalar value:
: 46 Text annotation                    :
: 47 Line drawing via cursor control     :
: 48 Digitize points                    :
: 49 Plot contours                      :
:      Type <ESC> to abort function.     :
:-----:
```

46

*Select the Text annotation function*

Enter type of font to use:

- 1) simplex
- 2) complex
- 3) italics
- 4) duplex

Enter font type ?

1

*Select the simplex font*

Enter text height ?

10

*Enter the font height in pixels*

Enter text rotation angle ?

0

*Text rotation in degrees ccw*

Enter value to use (0-255) ?

255

*Enter the VLT value to use in drawing (white)*

Enter text to annotate, then locate  
cursor at window position to annotate  
and press mouse button. (<ESC> to quit)

x label                      *Enter the text to use and press return*

Place the cursor at the lower left hand corner where the text will be placed and press any mouse button. The text will be drawn at that location. **Note:** pressing the mouse button again will draw the text where the mouse is located. The user may enter new text in the menu window without returning to the main menu and entering the font information again. This can be continued until the user presses the escape key. The length of the text is limited to 41 characters or the width of the menu window. Each subsequent click of the mouse will cause the current text to be written at the cursor location. When new text is entered, it is not put into the write buffer until a <CR> is entered. **Note:** there is no way to remove text from a window without clearing the entire window.

## 9 Basic Output

SDPS supports two basic types of output, SDPS format and PostScript format. The SDPS format is primarily used for raster-based output while PostScript is generally used for line-based output. SDPS can only generate images of line-based data in SDPS format. SDPS will create a PostScript format file that can be lines or images or both, but is primarily intended to print line drawings on a laser printer.

### 9.1 SDPS Format

This output is used to dump raster-based images from a display window. These images may be loaded later and may be used in most of the external routines (section 12). Two options are available for dumping images from a display window. Option 15 saves the data values in the bitmap including line overlays without CCLM modifications. Option 16 should be used when the changes made to the VLT and should be saved. This option only works for black and white images. To save a remapped color image, the user should first remap the color image using CCLM and then write down the cursor location given in the cursor window that produces the desired result. The user then selects a black and white VLT (option

21) and remaps the image using CCLM and with cursor location previously written down. Option 16 may then be selected to output the correct image. Option 16 also requests a slope and an intercept. This is optional and simply allows the user to add or change these values in the output header (section 5.4).

## **9.2 PostScript**

PostScript may also be used to generate images such as the ones printed in this manual, but with the availability of color hardcopy it is generally used for line-based drawings on a laser printer. Unlike raster output, PostScript uses the higher resolution of the laser printer to print straight lines and smooth curves. Therefore, if you are doing line operations in SDPS, you will want to use the PostScript output option. Option 74 allows the user to generate an image and overlay lines in PostScript format. Option 75 is used when the user wants to create a file containing only the lines.

## **9.3 UNIX**

SDPS also allows the user to output line-based data in standard UNIX plot format. This format is described in detail in the UNIX user manuals.

# **10 Advanced Functions**

SDPS provides functions for deriving additional information from a given data set. Using SDPS, one can modify a window's VLT to highlight subtle features in an image. The user can also sequentially display up to 15 images to detect time dependent phenomena. SDPS will display data along a line such as a ship track or a satellite ground track, overlay bitmaps to mask an image or draw freehand in a window. The user can also digitize points in an image, such as buoy or drifter locations, to an output file. SDPS also provides support for displaying images on workstations with monochrome screens. The examples given in this section are less detailed than the basic functions section and assumes the user has some familiarity with SDPS.

## 10.1 VLT Modification

When SDPS initially opens a window, a black and white (greyscale) VLT is automatically loaded in the window (option 21). This VLT is filled from 0 to 255 and when the cursor enters the display window the foreground and background colors used to display text in other windows are temporarily overwritten. This allows the user to use all available colors for displaying an image. Sometimes, the user may want to see the text in external windows (such as the cursor window) or may want to specify certain colors for lines or masks. This can be done by using option 28 to set the image truncation option to ON. When this option is turned ON, all images loaded will be truncated to 253 values instead of 255. Also, all subsequent VLTs that are loaded will be truncated and location 254 will be loaded with black (0,0,0) and 255 with white (255,255,255). **Note:** this option does not appear to take effect until a new VLT is loaded (options 21 and 22). If the user desires false color, SDPS supplies 3 distinct false color VLTs. One contains four colors, a second has ten colors, and the third is a 256 color rainbow. If any of these VLTs does not suit the user, a customized VLT may be loaded (option 24). The current VLT may also be customized with option 23 to change 1 to 256 consecutive colors. The entire VLT may also be customized by using *Cursor Controlled Linear Mapping* (CCLM option 26).

The truncation option may be a source of VLT problems. When the truncation option is set, all subsequent images are truncated. This means that values of 254 and 255 in the image will be set to 253. If the user wants to have a truncated VLT but not a truncated image first turn the truncation option on, next load in a VLT and turn the truncation option off, finally, load in the image file.

## 10.2 Movie Loop

SDPS may be used to display a sequence of up to 15 images in a time series. Images do not have to be the same size, however, all images are loaded into the upper left hand corner of the selected window and must be in SDPS format. The location of the upper left hand corner cannot be set for individual frames. The following example shows how to load and display five 512 by 512 images with 0.2 seconds separating each frame. The five files are named file.0, file.1, file.2, file.3 and file.4. The suffix must be number 0, 1, 2 etc., since SDPS uses this to read the files sequentially. This example assumes that a 512 by 512 display window has been opened.



30 *Select the Raster/vector image display submenu*

34 *Select the Display multiple SDPS image files function*

Enter filename of SDPS images

without # suffix.

(ie. 'frame.' for frame.0, .1, .2, ...)

file. *Enter the base filename of the images*

Enter starting memory # to load ?

1 *Start loading in memory location 1*

FILE I/O ERROR - file not read

*Known bug. Can usually be ignored.*

13 *This option controls the movie loop*

Enter window # to use ?

1 *Select an open window for the display*

Enter indices of images ( $\leq 15$ ) ?

1 2 3 4 5 *Order entered is order displayed*

Enter delay time (sec) ?

0.2 *Select 0.2 seconds between each frame.*

Enter number of cycles ?

5 *Enter number of times to repeat (delay is doubled between repeat cycles,*

Minimum delay time is limited by the type of machine running SDPS, system load and image size. Speed is also limited because the program does not use hardware or software buffering to keep the program portable.

### 10.3 Plotting Scalar Values Along a Track

SDPS has an option to plot a scalar value (salinity, sea surface height) along a line such as a ship or satellite ground track. To use this feature, the user must have entered the navigation information using option 41. Assuming the user has a file in SDPS format with the first column containing the latitude, the second the longitude and the third the desired value to plot, the following could be used to plot the data:

45                      *Select the Plot Lat/Lon track with scalar value function*

Enter window # to use ?

1                      *Select the appropriate window*

Enter SDPS input file name ?

file.sdps              *Enter the file name*

Enter indices of Lat,Lon,data ?

1 2 3                  *Select the correct columns*

Enter max expected data value ?

1.0                    *The data will be scaled +/- 1.0 units*

Enter height from track to display  
maximum data value ?

20                      *A data value of 1.0 will be scaled 20 pixels high*

Enter value for track, plot (0-255) ?

255 255                *Both the track and the data will be plotted using color 255*

An example of this is shown in figure 5.

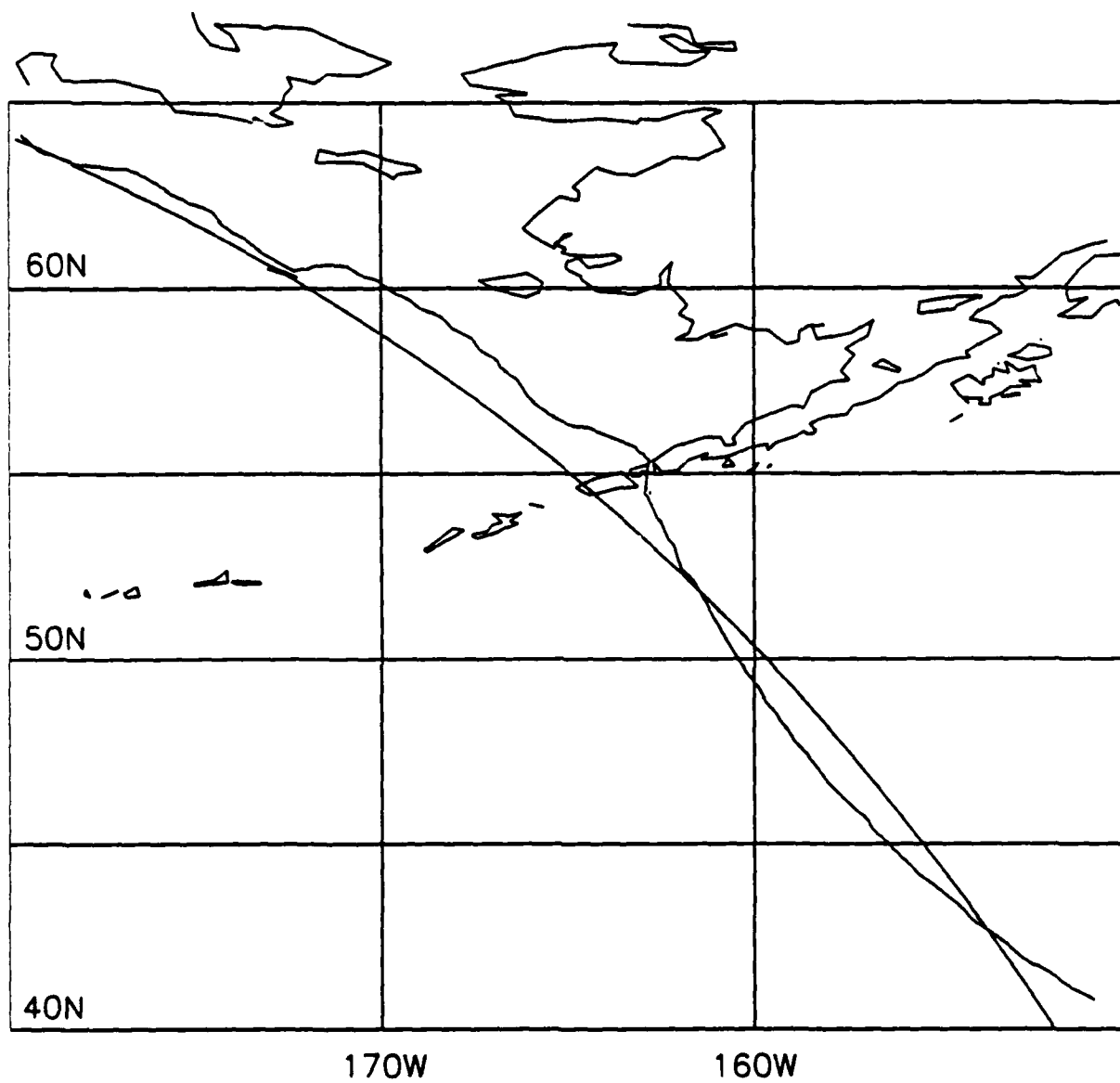


Figure 5: Example of a scalar plot. Coastline was added as a discontinuous line using option 43.

## 10.4 Line Drawing

SDPS provides several options for line drawing. All data files must be in SDPS format to be used in the plotting routines.

*Plot Navigation Lines* (option 42) allows the user to draw a latitude/longitude grid in a window. The user may draw latitude lines, longitude lines or both. The program requires a start value, a stop value and the increment between grid lines. Examples of this are shown in figures 3, 4 and 5.

*Plot Lat/Lon track* (option 43) is used for most line plotting. This function may be used to draw a continuous line, a discontinuous line or a set of points. The option for drawing a continuous line draws with the pen down line segments between each point in the file. For plotting continuous lines, the user may specify that the input data contain a series of continuous lines each containing the same number of points or separated by a flag value. Finally, each point of a data set may be plotted in the window. These points are plotted using + symbols and the size specified is the number of pixels high and wide.

*Plot vectors* (option 44) is used to plot arrow vectors. The data may be in the form of a magnitude and a direction or in u and v components. The u and positive 90 degree direction are assumed to be to the right. The vectors may be plotted in a single color where the length of the vector is plotted relative to the magnitude of the data point or where the color of the vector is relative to the magnitude of the data point.

*Plot Lat/Lon track with scalar value* (option 45) is used for plotting information relative to a lat/lon track.

*Line drawing via cursor control* (option 47) is used for drawing lines in a display window freehand. Select the first point of the line with the left mouse button and the second point with the middle mouse button. This point may be changed until the right mouse button is pressed to draw the connecting line.

## 10.5 Bitmap overlays

Options 35 and 36 allow the user to overlay bitmaps in an open display window. Option 35 reads an SDPS format file and performs a logical *AND* on the image previously loaded in the window. All zero values in the mask file will be set to zero in the image. All other values in the mask image will not affect the image displayed. Option 36 is similar except that a value of zero in the mask image will not affect the image displayed. All other values in the bitmap image will replace

the corresponding values in the display. The options are useful for masking out land values.

## **10.6 Digitize Points**

When a display window has a feature such as a front that the user would like to digitize, SDPS will digitize the cursor location and output the information to a file. If the navigation information has been set, the output values will be in latitude-longitude coordinates.

## **10.7 Dithered Images**

For users with monochrome workstations, most line-based functions will work. When an option asks for a VLT color, select 0 or 255 for black or white. Image and VLT functions will not work however, images may be displayed using the dithering options. Binary image files and SDPS format files can be displayed on a monochrome monitor using options 71 and 72. These functions display the image using a 4 by 4 dither to provide 17 different patterns (including black and white). Input parameters are the same as for options 32 and 33.

## **10.8 Defaults**

Many functions include default or commonly used values. These values are listed in square brackets [ ] where the default is available. To select the default values, simply press return at the prompt.

# **11 Menu Listings and Descriptions**

## **11.1 Window Functions**

### **11 OPEN WINDOW**

This function opens a window for displaying raster images or line plots. The size of the window is specified (in pixels) by the user. Currently, a window should not be resized after it is opened, as this will adversely effect the operation of other functions on this window. Preferably, the user should select QUIT from the window and open a new window if a different size is needed.

## **12 REDISPLAY IMAGE**

This function displays the raster information previously stored in memory by the user. This may be an image stored using option 14 or a memory location chosen when a file was initially loaded.

## **13 FLICK BETWEEN IMAGES**

This function displays a sequence of images previously stored in memory by the user individually using option 14, 31, 32 or 33, or sequentially using option 34. The user is prompted for the window to use, the delay time in seconds between frames and the indices of the memory locations to use. Currently, up to fifteen frames may be displayed.

## **14 OUTPUT WINDOW TO MEMORY**

This function allows the user to store a display window in memory as a raster image. Typically this is used when non-raster data such as line plotting or text annotation is plotted and the user would like to save it as a raster image to redisplay at a later time. This is also useful when a window is needed for an additional display or when the user wants to insert data into specific memory locations for flickering using option 13.

## **15 OUTPUT WINDOW TO FILE**

This function saves the data displayed in a window to a file as a raster image. The image file contains the actual data values in the bitmap. Changes in the VLT such as those done by CCLM (Cursor Controlled Linear Mapping) will not effect the saved image. Changes made to the bitmap such as notation and lines will be saved. To save changes made to the displayed intensities refer to option 16.

## **16 OUTPUT B&W REMAPPED WINDOW TO FILE**

This function saves the intensity values displayed in a window in SDPS file format. This function differs from the previous one because it uses the current VLT of the window to remap the data before it is output. This allows the user to display a black and white image, change the VLT with CCLM (Cursor Controlled Linear Mapping) and save the enhanced image. This option prompts for a slope and an intercept for the output image. Where *intercept* = (*true value*) corresponding to a screen byte value of zero and the *slope* = (*true value* - *intercept*) / 255. By just pressing return at this prompt, you will keep the current value of the window.

## **17 CLEAR WINDOW**

This function clears the user-specified window by entering zero (0) into all the pixels in the window bitmap. It does not clear any corresponding image in memory. The default value for the window number will be the last window accessed.

## **18 DISPLAY COLORBAR**

This function displays a colorbar for a specified window. If the image in that window was loaded as an SDPS format file and the header for that file has the slope and intercept set (section 5.4 ), the true values will be displayed along the colorbar. **Note:** that the colorbar displays colors in the current VLT.

## **19 DUMP WINDOW TO UNIX PLOT FILE**

This function allows the user to output the lines displayed in a window to a unix plot file <sup>6</sup>. This file may then be displayed with any routine used to display standard unix plot files.

# **11.2 VLT Manipulation**

## **21 LOAD BLACK/WHITE VLT**

This function loads a grey scale VLT (Video Lookup Table) into the user specified window. The VLT is a linear function with a slope of 1 and an intercept of 0. The VLT is pure black (0, 0, 0) for pixel values of 0 and pure white for pixel values of 255. If the auto-truncation option is on, pure white (255, 255, 255) is truncated at 253 and 254 is reserved for black while 255 is reserved for white.

## **22 LOAD COLOR VLT**

This function loads a pseudo-color VLT for the user specified window. The user has a choice of a 4, 10, or 256 color VLT. The four color VLT provides four distinct colors: blue, green, yellow or red. The ten color VLT provides the following distinct colors: purple, midnight blue, dark blue, light blue, dark green, light green, yellow, orange, red and dark red. The 256 color VLT provides the same basic colors as the 10 color VLT, but also provides intermediate shades.

---

<sup>6</sup>See UNIX manual for more information on plot file format

### **23 LOAD INDIVIDUAL VLT VALUES**

This function allows the user to change one or more VLT values to any user specified RGB value. This is useful when a user wishes to highlight a feature that is represented by a single value, such as to give navigation lines and coastlines different colors when plotting. The values for red, green and blue should range from 0, 0, 0 (black) to 255, 255, 255 (white). The user must supply the VLT starting position and the number of values to change. These values should vary from 0 to 255 and the total should be less than 256. The user should have some familiarity with RGB color concepts.

### **24 READ VLT FROM FILE**

This function reads a binary file of  $256 \times 3$  byte values to specify the VLT of a window. SDPS does not support ASCII files. This function is primarily intended to read a VLT written by the option 25. The user may create a VLT by writing a binary file with 256 byte values for red followed by 256 byte values for green and 256 byte values for blue.

### **25 WRITE VLT TO FILE**

This function writes the VLT from a user specified window to a file. This file may be read later using option 24. This is useful for saving VLT modified using option 26.

### **26 CURSOR CONTROLLED LINEAR MAPPING**

This function uses the cursor to control a linear VLT mapping function. The cursor position is read and the VLT for the window specified is remapped to another VLT which is then used for display. The vertical position of the cursor determines the slope of the function and the horizontal position determines intercept. Hence with the cursor in the middle of the window the slope is zero, at the ends it is infinity, and three quarters up from the bottom it is one, and one quarter up from the bottom it is negative one. With the cursor in the middle an old VLT value of 128 is mapped to 128. With the cursor to the left it is mapped to 0, and with the cursor to the right it is mapped to 255. Note: the entire image is used for the mapping not just the view window.

### **27 INVERT VLT VALUES**

This function inverts the values of the VLT for a window so that the color for 0 is now used for 255, and the color for 255 is now used for 0. This changes



the sign of the slope of the mapping function but not the magnitude. If the truncation option is set, only the first 253 values are inverted.

## **28 SET AUTO-TRUNCATION OPTION**

This option when set to **ON**, will automatically truncate values of input raster images to the 0-253 range, setting values of 255 and 254 to 253. Also, the VLTs that are placed into windows or assigned at window creation will reserve 254 and 255 for black and white respectively. This allows the other **SUNVIEW** windows to be viewed simultaneously with the display windows. To preserve values of 254 and 255, set auto-truncation **OFF** and then load the image. If black and white values are needed for the VLT locations 254 and 255, then turn auto-truncation **ON** and load in a VLT. This will give an untruncated image with a truncated VLT. This function cannot be specified separately for each window. Once specified it will remain in effect for all windows until turned **OFF**. VLTs will not be affected by this option until a new VLT is loaded.

## **29 SET CURSOR COLOR**

This option allows the user to change the color of the cursor. It is sometimes necessary to change the cursor color in a window to distinguish it from the image. The byte value (from 0 to 255) is set to the corresponding color in the window's VLT.

# **11.3 Raster Image Display**

## **31 DISPLAY SUN RASTERFILE**

This function reads a **SUN** rasterfile, which contains a header, an optional VLT and an image, and displays it in the user specified window. The last window opened is both the default window number and the default memory number. The default display position is row zero and column zero which corresponds to the upper left hand corner.

## **32 DISPLAY BINARY IMAGE FILE**

This function reads a file containing an optional header and a byte image of any size and displays the image in the specified window. Here, the default memory number, window number and display position are the same as for option 31. The default image size corresponds to the image size used when opening the window (see option 11). The default number of header bytes to

skip is zero. If header bytes are not skipped, they will be displayed as part of the image.

### **33 DISPLAY SDPS IMAGE FILE**

This function reads a file in SDPS file format and displays the 2D data as a raster image in the specified window. This function currently only reads byte files. For information on converting file formats see sections 5 and 12.2. The defaults used here for memory number, window number and display position are the same as for options 31 and 32.

### **34 READ MULTIPLE SDPS IMAGE FILES**

This function reads multiple SDPS format files into memory to be displayed later. These files must have .0, .1, .2, ... as suffixes in order to be read in correct order. This function is typically used with multiple files which are viewed with option 13 in a movie-loop fashion. The images are stored in consecutive memory positions starting at the specified memory position.

### **35 DISPLAY SDPS MASK FILE**

This function reads an SDPS file and uses it to mask an image previously loaded in a window. This is done by performing a logical *AND* of the image file read in with the image in the display window. Locations with 0 in the mask file will be set to zero, while values of 255 will not affect the image in the display window.

### **36 OVERLAY SDPS BIT MAP**

This function is similar to 35 except that it places the value of the bitmap file in the window for all values greater than 0. A bitmap value of 0 will not be overlaid on the window.

## **11.4 Track Data Plotting**

### **41 SET NAVIGATION AREA**

This function sets the coordinate system for line plotting in the display windows and must be used before any plotting is attempted. There are three map projections to choose from: *Stereographic*, *Mercator*, and *Equirectangular*. Each projection has three parameters to describe the size and orientation. The first is center latitude and longitude, which defines the latitude and longitude at the point in the center of the window. Second is rotation,

which defines rotation of the coordinate system about the center point in the window. Thirdly, x and y scale factors determine the scale in the horizontal and vertical direction of the window.

For the *Stereographic* projection, a scale of one in each direction allows the zero latitude line to fit just within the boundaries of the window. With a scale of one for the *Mercator* projection, 360 degrees of longitude can fit horizontally in the window and just above and below positive and negative 80 degrees latitude. The *Equirectangular* projection allows an even proportion of degrees per distance in both the latitude and longitude directions, and so 360 degrees will fit both horizontally and vertically in the window with x and y scale factors of one.

#### 42 DRAW NAVIGATION LINES

This function allows the user to draw latitude and longitude lines in a specified window given a minimum, maximum and incremental value for latitude and longitude. The user may chose a VLT value between 0 and 255 to use for writing to the window. The parameters in option 41 must be set before using this option.

#### 43 PLOT TRACK

This function reads an SDPS format file containing latitude and longitude coordinates. The input data may represent a continuous line, a discontinuous line or a set of individual points. If the continuous line option is chosen, the program will draw all points connected with a single continuous line. It will attempt to break the line if it crosses the window boundary. There are two types of discontinuous lines available. The user may input a data set that is broken by a specified delimiter or a data set that contains segments of equal length. The user may also choose to input a data set that contains discrete points. The program plots these points using a small plus sign. The data is assumed to be in columnar format where each row contains a latitude coordinate, a longitude coordinate and optionally any other data such as the magnitude and direction to be used in a vector plot. In order to use an ASCII file for this option, the user must first convert to SDPS format using option 51. This converts the file to binary for faster reading/writing. The user may choose a VLT value between 0 and 255 to use for writing to the window.

#### **44 PLOT VECTORS**

This function reads an SDPS format file and plots magnitude and direction in the form of variable length arrows. The positional coordinates must be latitude and longitude. The vector may be defined as x and y components or as magnitude and direction. The user may chose a value between 0 and 255 to use for writing to the window. X is assumed to be positive to the right and y is positive to the top. Direction is given as positive 90 degrees to the right and 0 degrees to the top.

#### **45 PLOT SCALAR VALUES ALONG TRACK**

This function reads an SDPS format file and plots a latitude, longitude track and a line with the x-axis parallel to the track and the y-axis perpendicular to the track. The user is then asked for a maximum expected value. A positive value is interpreted as the absolute value of the data and is used to scale the data. If a negative value is selected, the program finds the maximum absolute value of the data, uses it to scale all the data and prints the maximum on the screen. If this value is less than the maximum encountered by the program, the actual maximum will be displayed. The user is then prompted for the height from the track to display maximum value, i.e., the number of pixels between the latitude-longitude track and the maximum value to be plotted. The user then choses separate values between 0 and 255 for plotting the latitude-longitude track and the line.

#### **46 TEXT ANNOTATION**

This function reads an input character string and outputs it to a user specified location in a display window.

#### **47 LINE DRAWING VIA CURSOR CONTROL**

This function allows the user to input straight lines to a window using the cursor. Clicking the left mouse button locates the starting point, the middle mouse button locates the end point, and clicking right mouse button draws the line. The left and middle mouse buttons echo their position when pressed.

#### **48 ECHO CURSOR LOCATION**

This function prints the cursor position in a display window when the left mouse button is clicked. This function prints the cursor values to the menu window and to a file specified by the user. If option 41 has set the navigation

parameters, the output will be in lat-lon coordinates. Otherwise, the row and column values are printed.

#### **49 PLOT CONTOURS**

This function plots selected contours from an SDPS format file. The file must be rectangular, but does not need to have even spacing. The user may specify a mask file for areas that should not be contoured. This file is optional and if it is not specified, the default is to use all data points in the input file. The latitude and longitude are input with one dimensional SDPS format files. Each entry in the longitude file correspond to each data point from left to right and each entry in the latitude file correspond to each data point from top to bottom. The user may select a range of contour levels and the increment between each level. The user may also select a range of colors to plot each contour level from 0 to 255. To select a single color, enter the number twice. **Note:** The input file, the longitude and latitude files are floating point format files and the mask file is a byte type file.

### **11.5 File Reformatting**

#### **51 REFORMAT ASCII TO SDPS**

This function reads an ASCII file with a fixed number of columns of data and converts it to an SDPS format file of two dimensions. A more flexible and robust method of converting files from ASCII to SDPS is done in the external routine *sdps\_convert* (section 12.2).

#### **52 REFORMAT BINARY IMAGE TO SDPS**

This function read a binary image file and strips off header information, then stores it as an SDPS format file of two dimensions. This option does not set the slope and intercept values. These may be set in the external subroutine *sdps\_headedit* (section 12.7).

### **11.6 Data Processing Functions**

#### **61 CONVERT EAST/NORTH TO LAT./LON.**

This function reads a format file containing N columns of data with two columns of east and north distances in kilometers, and using a basepoint latitude, longitude position, changes the position data to latitude, longitude

coordinates. The result is then written to an SDPS format file of the same size and number of columns. This option may be removed in future versions.

## **11.7 Dither And PostScript Functions**

### **71 DISPLAY DITHERED IMAGE**

This function is similar to option 32. It is used to display a binary image file on a monochrome monitor. The image is displayed by dithering a 4 by 4 pixel area using 17 different patterns (including black and white). The average of the 4 by 4 area is used to determine which dither pattern to display. Since some SDPS functions do not work on monochrome monitors, images loaded using this function are useful only for previewing images.

### **72 DISPLAY DITHERED SDPS**

This function is similar to option 71 except that it dithers SDPS format files.

### **73 DISPLAY DITHER BAR**

This function opens a new window and displays the different dither patterns. It does not display the pixel values like option 18.

### **74 DUMP PS SDPS IMAGE**

This function reads an SDPS format file from disk and formats the output in a PostScript-compatible file. This function optionally overlay lines from a display window. The image file is read from disk and not the window so that overlaid lines will not be dithered with the image. There is also an optional colorbar that displays the values associated with ten different greyscales. These values will only be displayed correctly if the input file has the slope and intercept set in the header (section 5.4).

### **75 DUMP LINES TO PS FILE**

This function reads the lines and vectors from a display window and writes them to a PostScript file. This is an ASCII file and may be edited for special effects or may be sent directly to a PostScript printer.

## **11.8 Image Functions**

### **81 MAGNIFY IMAGE**

This function allows the user to read an image from one window, magnify

it and put it in a second window. The user may choose from two types of magnification: simple pixel replication and bi-linear interpolation. The simple pixel replication technique allows the user to specify a magnification factor in each of the row and column directions. The bi-linear interpolation technique only accepts magnification factors that are powers of 2 and does not allow for separate row and column factors. The bi-linear technique will produce a smoother magnified image, but will also blur sharp features such as coastlines.

## 82 MATH FUNCTIONS

This function allows the user to add, subtract, multiply or divide one image from/by another. The result is placed in the second window. The images loaded in the two display windows must be in SDPS format and must have the slope and intercept specified. This option converts back to real before performing the operation and then converts back to byte before loading the results back into the display window. All operations are done on the image in the first window by the image in the second window and are displayed in the second window. For example:  $image2(i,j) = image1(i,j) - image2(i,j)$ <sup>7</sup>.

## 83 HISTOGRAM

This function opens a new window and displays the histogram of a specified display window. The color of each bar corresponds to the same color pixels in the display window. **Note:** this function only works with color monitors and is not automatically updated when the window contents change.

# 12 External Utilities

This section describes various utilities that may be used to perform image manipulation without invoking SDPS.

## 12.1 sdps\_append

This program is used to append two SDPS format files together. Usage:

```
sdps_append row_off col_off file1.sdps file2.sdps > file3.sdps
```

---

<sup>7</sup>See section 5.4 for more information on SDPS file types

The first file is located at 0,0 and the second file is located at integer location *row\_off*, *col\_off* as specified on the command line. *Col\_off* is the offset along a row and *row\_off* is the offset along a column. *File3.sdps* is a new SDPS format file that is *col\_off* plus the number of columns in *file2.sdps* wide and *row\_off* plus the number of rows in *file2.sdps* high. **Note:** that the upper left hand corner of *file2.sdps* and *file3.sdps* is located at 0, 0. Therefore if *file1.sdps* has 200 rows and 300 columns and *file2.sdps* has 300 rows and 100 columns, the following command would be used to append *file2.sdps* to the right of *file1.sdps*:

```
sdps_append 299 0 file1.sdps file2.sdps > file3.sdps
```

while the following command appends *file2.sdps* to the bottom of *file1.sdps*:

```
sdps_append 0 199 file1.sdps file2.sdps > file3.sdps
```

*File3.sdps* would have 300 rows and 400 columns in the first example and 500 rows and 300 columns. Any location not filled from *file1.sdps* or *file2.sdps* would be filled with a 0.

## 12.2 sdps\_convert

This program is used to convert ASCII files to SDPS floating point files. Usage:

```
sdps_convert rows cols [drow dcol] < file1.asc > file2.sdpsf
```

The ASCII file is assumed to be free format with at least *rows* times *cols* data points. The program checks for an end-of-file so that the argument *rows* may be larger than the actual number of rows in the input file. The program also assumes the data to be rectangular and ordered from left to right, top to bottom with the number of columns known. The user may optionally enter the row and column spacing using the input parameters *drow* and *dcol*. These values are used in other auxiliary programs. For more information on *drow* and *dcol* see SDPS format files (section 5.4). For example, if *file1.sdps* contains the following ASCII latitude-longitude ship locations:



```
10.0 290.0
10.25 292.35
10.51 293.61
```

the following command would be used to convert line data to an SDPS float format file:

```
sdps_convert 3 2 < file1.asc > file2.sdpsf
```

*File2.sdpsf* could now be used in SDPS to identify the ship tracks.

### 12.3 sdps\_cut

This program cuts a section from an SDPS format file. Usage:

```
sdps_cut rowul colul rowlr collr < file1.sdps > file2.sdps
```

The program reads *file1.sdps*, cuts the section from *rowul*, *colul* to *rowlr*, *collr* and writes the resulting *file2*. The program keeps the original slope and intercept header values from *file1.sdps* so that *file2* will use the same scale. The only modification to the output header file is to change the size. **Note:** the upper left corner of the data file is located at 0, 0. To cut the upper left 256 by 256 section from a 512 by 512 file, the following command would be used:

```
sdps_cut 0 0 255 255 < file1.sdps > file2.sdps
```

### 12.4 sdps\_flip

This program flips an SDPS file around a row or column axis. Usage:

```
sdps_flip dir < file1.sdps > file2.sdps
```

If *dir* is 1, the program flips the file around the image from top to bottom. If *dir* is 2, the program flips the file around from left to right. For example:

```
sdps_flip 1 < file1.sdps > file2.sdps
```

Makes the top row of *file1.sdps* the bottom row in *file2.sdps*.

## 12.5 sdps\_ftb

This subroutine reads SDPS float format files and converts them to byte format files. Usage:

```
sdps_ftb [-mxlh] [min max low high] < file1.sdpsf > file2.sdps
```

Each point is scaled between the *max* and *min* to the range of byte values (0-253 for truncated and 0-255 for full files). The options specify how to perform the scaling:

Option	Description
-m	Use the user specified value as minimum for scaling.
-x	Use the user specified value as maximum for scaling.
-l	Values < <i>min</i> are specified by user (use with -m)
-h	Values > <i>max</i> are specified by user (use with -x)
-f	Use all 256 byte values for scaling the output

The values for *min* and *max* are floating point and *low* and *high* are integer (0 - 255). **Note:** that the order of the options must correspond to the order of the variables and the options must precede the variables: *-xm max min* is acceptable, but *-m min -x max* is not. By default, the output file values are truncated (0 - 253) with 254 and 255 reserved for black and white.

To create a truncated byte image with 0 as the minimum and 253 as the maximum of the floating point image:

```
sdps_ftb < file1.sdpsf > file2.sdps
```

*File2.sdps* will be an SDPS byte image.

If a floating point file has a -999.0 mask flag, a maximum of 20.0, minimum of -10.0 and the user wants to mask the -999.0 values with black on the byte image file:

```
sdps_ftb -mxlh -10.0 20.0 254 254 < file1.sdpsf > file2.sdps.
```

*File2.sdps* will be scaled to a minimum of -10.0 and a maximum of 20.0. All values less than -10.0 or greater than 20.0 will be set to 254 (black)

## 12.6 sdps\_header

This program displays the header from an SDPS format file. Usage:

```
sdps_header < file1.sdps
```

## 12.7 sdps\_headedit

This program edits the header from an SDPS format file. Usage:

```
sdps_headedit file1.sdps > file2.sdps
```

The program displays the current value of each header value and queries the user for a new value. Pressing return retains the old values.

## 12.8 sdps\_logscale

This program rescales an SDPS float file to a log scale. Usage:

```
sdps_logscale < file1.sdpsf > file2.sdpsf
```

This program amplifies an SDPS float format file (such as a magnitude file). The program assumes that the input file only contains positive numbers. If negative numbers are found, the program prints an error message and exits. This program only works on SDPS float format files.

## 12.9 sdps\_mask

This program masks one SDPS format file with another. Usage:

```
sdps_mask file1.sdpsf file2.sdps threshold [mask_value] > file3.sdpsf
```

The program reads *file1.sdpsf* as the input file to be masked and *file2.sdps* as the mask file. *Threshold* is the value to compare against the mask file. If the mask file is less than the threshold value, then the original point in *file1.sdps* is left intact. If the value in the mask file is greater than or equal to the threshold value, the original point is changed to zero or to the optional *mask\_value*. This program copies the header from *file1.sdps* to *file3.sdpsf* and does not modify the slope or the intercept. The program masks the section beginning in the upper left hand corner of the original image. The mask image must be less than or equal to the size of the image to be masked. *File1.sdps* is a float type file, *file2.sdps* is a byte type file, *threshold* is an integer and *mask\_value* is a floating point value.

## 12.10 sdps\_math

This program adds, subtracts, multiplies or divides two SDPS format files together. Usage:

```
sdps_math op file1.sdpsf file2.sdpsf > file3.sdpsf
```

The math operation *op* is performed by *file2.sdpsf* on *file1.sdpsf* and the result is placed in *file3.sdpsf*. The option *op* indicates which operation to perform:

Option	Operation
a	Add file2.sdpsf to file1.sdpsf.
s	Subtract file2.sdpsf from file1.sdpsf.
m	Multiply file1.sdpsf with file2.sdpsf.
d	Divide file1.sdpsf by file2.sdpsf.

## 12.11 sdps\_pad

This program pads one or more edges of an SDPS format file with a specified value or with the average value of the image. Usage:

```
sdps_pad flags pad_size pad_val < file1.sdpsf > file2.sdpsf
```

This program places the value given in *pad\_val* around the edges specified in the *flags* argument. The number of points added to each edge is the integer value *pad\_size*. If *pad\_val* is an ASCII character, the mean of the image will be used for padding. The argument *flags* is used to specify how to pad the image:

Option	Description
l	Pads left edge of image.
r	Pads right edge of image.
t	Pads top edge of image.
b	Pads bottom edge of image.
a	Pads all edges of image.

Flags can be combined to pad more than one edge. The flags *lrtb* are the same as the single flag *a*. **Note:** that flag *a* should be used alone. To pad an SDPS format file on the top and right edge with the average value of the image:

```
sdps_pad tr 1 a < file1.sdpsf > file2.sdpsf
```

To pad this image with 2 pixels of 0.0 around all edges:

```
sdps_pad a 2 0.0 < file1.sdpsf > file2.sdpsf
```

## 12.12 sdps\_rotate

This program rotates an SDPS format file around the origin. Usage:

```
sdps_rotate deg < file1.sdps > file2.sdps
```

The command line argument *deg* must be 90, 180 or 270. A positive number indicates rotation in the counter-clockwise direction, while a negative number indicates rotation in the clockwise-direction. To rotate an image 90 degrees counter clockwise:

```
sdps_rotate 90 < file1.sdps > file2.sdps
```

## 12.13 sdps\_to\_sun

This program converts an SDPS format image file to a SUN rasterfile. Usage:

```
sdps_to_sun [-ac vltname] < file1.sdps > file2.sun
```

By default, it uses the same truncated full color VLT that SDPS uses. This assigns 254 to black and 255 to white. The user may optionally specify an ASCII VLT with values of red, green and blue on a separate line. The program expects to read 256 rows of colors. If these data are in file *vlt.asc*, then the following will use that VLT:

```
sdps_to_sun -a vlt.asc < file1.sdps > file2.sun
```

The user may also specify a binary VLT as output using option 25 in SDPS. If this VLT is in file *vlt.sdps*, then the following will convert an SDPS image to a SUN rasterfile:

```
sdps_to_sun -c vlt.sdps < file1.sdps > file2.sun
```

### **Acknowledgements**

The authors would like to thank Drs. Nick Fofonoff, Hans Graber and Kathryn Kelly for their support and advice in this project, Carol Alessi, Roger Goldsmith, and Debbie Barber for their suggestions on the content of this manual and Derek Spencer for providing additional funding for the Remote Sensing Analysis Laboratory.

## A Example ASCII Line data file

```
27.600000 286.670013
27.629999 287.779999
27.690001 288.579987
27.860001 289.309998
28.129999 290.070007
28.360001 290.720001
28.620001 291.359985
28.920000 291.920013
29.500000 292.510010
29.850000 292.829987
30.260000 293.209991
31.170000 293.649994
32.020000 294.179993
32.779999 294.480011
33.599998 294.890015
```

## B Example ASCII Vector data file

```
285.0 25.0 1.0 5.0
287.0 25.0 2.0 4.0
289.0 25.0 3.0 3.0
291.0 25.0 4.0 2.0
293.0 25.0 5.0 1.25
295.0 25.0 6.0 0.75
286.0 28.0 2.0 4.5
288.0 28.0 3.0 3.55
290.0 28.0 3.8 2.4
292.0 28.0 4.9 1.8
294.0 28.0 5.6 2.1
285.0 31.0 1.5 5.0
287.0 30.0 1.75 4.9
289.0 31.0 2.0 4.7
291.0 30.0 2.75 3.0
293.0 31.0 2.25 2.25
```

295.0 30.0 2.0 1.75  
285.0 34.0 1.5 5.3  
287.0 34.0 1.75 5.2  
290.0 34.0 2.0 5.1  
292.0 34.0 2.25 4.0  
295.0 34.0 2.5 4.0



## DOCUMENT LIBRARY

May 5, 1989

### *Distribution List for Technical Report Exchange*

Attn: Stella Sanchez-Wade  
Documents Section  
Scripps Institution of Oceanography  
Library, Mail Code C-075C  
La Jolla, CA 92093

Hancock Library of Biology &  
Oceanography  
Alan Hancock Laboratory  
University of Southern California  
University Park  
Los Angeles, CA 90089-0371

Gifts & Exchanges  
Library  
Bedford Institute of Oceanography  
P.O. Box 1006  
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International  
Ice Patrol  
c/o Coast Guard R & D Center  
Avery Point  
Groton, CT 06340

Library  
Physical Oceanographic Laboratory  
Nova University  
8000 N. Ocean Drive  
Dania, FL 33304

NOAA/NESDIS Miami Library Center  
4301 Rickenbacker Causeway  
Miami, FL 33149

Library  
Skidaway Institute of Oceanography  
P.O. Box 13687  
Savannah, GA 31416

Institute of Geophysics  
University of Hawaii  
Library Room 252  
2525 Correa Road  
Honolulu, HI 96822

Library  
Chesapeake Bay Institute  
4800 Atwell Road  
Shady Side, MD 20876

MIT Libraries  
Serial Journal Room 14E-210  
Cambridge, MA 02139

Director, Ralph M. Parsons Laboratory  
Room 48-311  
MIT  
Cambridge, MA 02139

Marine Resources Information Center  
Building E38-320  
MIT  
Cambridge, MA 02139

Library  
Lamont-Doherty Geological  
Observatory  
Columbia University  
Palisades, NY 10964

Library  
Serials Department  
Oregon State University  
Corvallis, OR 97331

Pell Marine Science Library  
University of Rhode Island  
Narragansett Bay Campus  
Narragansett, RI 02882

Working Collection  
Texas A&M University  
Dept. of Oceanography  
College Station, TX 77843

Library  
Virginia Institute of Marine Science  
Gloucester Point, VA 23062

Fisheries-Oceanography Library  
151 Oceanography Teaching Bldg.  
University of Washington  
Seattle, WA 98195

Library  
R.S.M.A.S.  
University of Miami  
4600 Rickenbacker Causeway  
Miami, FL 33149

Maury Oceanographic Library  
Naval Oceanographic Office  
Bay St. Louis  
NSTL, MS 39522-5001

Marine Sciences Collection  
Mayaguez Campus Library  
University of Puerto Rico  
Mayaguez, Puerto Rico 00708

<b>REPORT DOCUMENTATION PAGE</b>	<b>1. REPORT NO.</b> WHOI-89-13	<b>2.</b>	<b>3. Recipient's Accession No.</b>
<b>4. Title and Subtitle</b> Satellite Data Processing System (SDPS) Users Manual V1.0			<b>5. Report Date</b> May 1989
			<b>6.</b>
<b>7. Author(s)</b> Michael Caruso, Chris Dunn			<b>8. Performing Organization Rept. No.</b> WHOI-89-13
<b>9. Performing Organization Name and Address</b>  The Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543			<b>10. Project/Task/Work Unit No.</b>
			<b>11. Contract(C) or Grant(G) No.</b> (C) N00014-86-K-0751; 957652 (G)
<b>12. Sponsoring Organization Name and Address</b> The Office of Naval Research; and the National Aeronautic and Space Administration			<b>13. Type of Report &amp; Period Covered</b> Technical Report
			<b>14.</b>
<b>15. Supplementary Notes</b> This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-89-13.			
<b>16. Abstract (Limit: 200 words)</b> SDPS was written to allow users to quickly and easily display image and line-based data on inexpensive Sun workstations. The primary purpose of the program is for assimilating oceanographic data that may be generated from a satellite, a ship or a buoy. This manual describes the various options available and examples of how to use them. Also discussed are several additional programs for managing data external to SDPS.			
<b>17. Document Analysis</b>			
<b>a. Descriptors</b>			
1. satellite			
2. graphics			
3. image processing			
<b>b. Identifiers/Open-Ended Terms</b>			
<b>c. COSATI Field/Group</b>			
<b>18. Availability Statement</b> Approved for publication; distribution unlimited.		<b>19. Security Class (This Report)</b> UNCLASSIFIED	<b>21. No. of Pages</b> 58
		<b>20. Security Class (This Page)</b>	<b>22. Price</b>